Smart Computing Review

Comparative Study of Hu Moments and Zernike Moments in Object Recognition

Reza Kasyauqi Sabhara[1], Chin-Poo Lee[2], and Kian-Ming Lim[3]

[1]Faculty of Information Science and Technology, Multimedia University / Jalan Ayer Keroh Lama, Malacca, Malaysia / kasyauqi@gmail.com

[2] Faculty of Information Science and Technology, Multimedia University / Jalan Ayer Keroh Lama, Malacca, Malaysia / cplee@mmu.edu.my

[3] Faculty of Information Science and Technology, Multimedia University / Jalan Ayer Keroh Lama, Malacca, Malaysia / kmlim@mmu.edu.my

* Corresponding Author: Chin-Poo Lee

Received March 18, 2013; Revised May 17, 2013; Accepted May 24, 2013; Published June 30, 2013

Abstract: There are lots of ways to perform object recognition. This paper is part of a project studying object recognition. The project is intended as a starting point to further learning about object recognition. Therefore, moment invariants are studied as a good starting point. Hu moment invariant methods and Zernike moment invariant methods are implemented and compared. Zernike moment invariants are shown to outperform Hu moment invariants.

Keywords: Object recognition, moments, Hu moments, Zernike moments

Introduction

In just a few years following the inception of computers, scientists became fascinated by the possibility of building intelligent machines—machines that can think and behave like humans and become part of our lives. In 1950, Alan Turing, one of the fathers of artificial intelligence, suggested that the ability to understand the visual world is a prerequisite for such machines. In conjunction with this goal, many methods have been developed to allow machines to understand the visual world. One of the earliest and most used methods is moment invariants.

This paper is an effort to understand the vision systems that can recognize external objects through optical devices to enable a computer-vision approach, and was initiated by extensively studying two moment invariant types for image feature

DOI: 10.6029/smartcr.2013.03.003

extraction.

Moment invariants were chosen because they are one of the earliest methods employed to perform object recognition, and they have been continuously developed; thus they have significant history and influence on the object recognition field. Furthermore, they are one of the most important and most used methods in the field [2]. Therefore, studying this type of shape descriptor is a good starting point for providing an appropriate background in object recognition.

In this paper, Hu and Zernike moment invariants are implemented and compared. Hu moments were chosen because they are the earliest method that performed object recognition using moments. Zernike moments were chosen because they are one of the best descriptors in terms of overall performance, as explained by Teh and Chin [1].

Section 2 describes moments, including geometric moments, complex moments, and orthogonal moments. Section 3 explains Hu moments, with Zernike moments explained in Section 4. Section 5 discusses the image database, and Section 6 discusses ideas in object recognition. Results are shown in Section 7, and Section 8 concludes the paper.

Moments

Flusser et al. [2] define moments as scalar quantities used to characterize a function and to capture its significant features. Furthermore, they provide clearer and convenient definition of moments [2]:

Definition 1: By an image function (or image) we understand any piece-wise continuous real function $f(x, y)$ of two variables defined on a compact support $D \subset RxR$ and having a finite nonzero integral.

Definition 2: General or regular moment M_{pq} of an image $f(x,y)$, where p, q are non-negative integers and $r = p+q$, is called the order of the moment, defined as as the formula in Appendix (1), where $p_{pq}(x,y)$ are polynomial basis functions defined on D, and p,q are 0, 1, 2, 3... . When we say third order moment, that could mean m_{30}, m_{03}, m_{21}, or m_{12}. Depending on the polynomial basis used, there are various systems of moments.

Geometric and complex moments

If we use $p_{pq}(x,y) = x^k y^j$ as the polynomial basis, we will have geometric moments in Appendix (2):

Low-order geometric moments have their own unique meaning; for example:

- m_{00} is the mass of an image. For a binary image, it is the area of the object.
- m_{10}/m_{00} and m_{10}/m_{00} is the coordinate of the center of gravity, or centroid, of an image.
- m_{20} and m_{02} define the moments of inertia of an image.

Any image function has geometric moments of all orders and is finite. The image function can be reconstructed from the set of its moments [2]. An example of algorithms defined over geometric moments would be Hu moment invariants. If we use $P_{pq}(x,y) = (x + iy)^k(x-iy)^j$, where i is the imaginary units, we will have complex geometric moments in Appendix (3):

Geometric moments and complex moments carry the same amount of information. Complex moments are introduced because they behave favorably under image rotation [2].

Orthogonal moments

If the polynomial basis $p_{pq}(x,y)$ is orthogonal, i.e. if its elements satisfy the condition of orthogonality, the orthogonal moments will be the formula in Appendix (4) or weighted orthogonally as defined in Appendix (5).

For any indexes $p \neq m$ or $q \neq n$, and Ω as the area of orthogonality, then we have orthogonal (OG) moments. Unlike geometric moments, OG moments are coordinates of f on a polynomial basis in the sense commonly used in linear algebra. Thus, reconstruction of OG moments can be performed easily in the spatial domain. On the other hand, image reconstruction from geometric moments cannot be performed directly in the spatial domain. It is carried out in the Fourier domain [2].

Hu Moments

Hu (1962) defined seven moment invariants from geometric moments that are invariants to rotation. The seven features are shown in Appendix (6).

In order to make the features translation and scaling invariants, we need to substitute the geometric moment with the normalized central moment. The η_{pq} notations used above are called normalized central geometric moments. As the name suggests, in order to obtain these moments, we need to obtain the central moment, μ_{pq}, from the geometric moment and then compute the normalized moment, η_{pq}, from this central moment. Thus, we will obtain translation, scaling and rotation invariant features.

A two-dimensional ($p+q$)-th *order* general geometric moment of an $M \times N$ image is defined (in the discrete domain) as:

$$m_{pq} = \sum_{x=0}^{x=M-1} \sum_{y=0}^{y=N-1} (x)^p \cdot (y)^q f(x,y) \tag{1}$$

In a binary image, an area of an object, or the contour in this case, is held by m_{00}. The contour's centroid then can be calculated from:

$$\bar{x} = \frac{m_{10}}{m_{00}} \qquad \bar{y} = \frac{m_{01}}{m_{00}} \tag{2}$$

where x and y are points on the x and y axes, respectively. Using this centroid, we can transform the moment mentioned earlier into a *translation invariant* moment by redefining it into a *central moment*, defined as

$$\mu_{pq} = \sum_{x=0}^{x=M-1} \sum_{y=0}^{y=N-1} (x-\bar{x})^p \cdot (y-\bar{y})^q f(x,y) \tag{3}$$

A normalized central moment is *scale invariant*. Therefore the central moment can be transformed into a normalized moment by

$$\begin{gathered} \gamma = [(p+q)/2]+1 \\ \eta_{pq} = \mu_{pq} / \mu_{00}^{\gamma} \end{gathered} \tag{4}$$

Finally, to make the moment orientation (rotation) invariant, we simply use η_{pq} on Hu's seven features above instead of a standard geometric moment.

Hu described M_1 to M_6 as absolute orthogonal invariants (independent of position, size, and orientation) and M_7 as a skew orthogonal invariant (useful in distinguishing mirror images). These features are capable of recognizing simple objects, such as a character in Hu's experiment.

To provide a proof of the rotation invariance, two of Hu's moments from human contour images, shown in Figure 1, are calculated. In Figure 1 on the right is a 90-degree rotation of the figure on the left. From Table 1 and Figure 1, Hu's moments are perfectly equal for M_1 to M_6. M_7 is different because it is a feature of a skew invariant, whereas the rest are used for position, size and rotation invariants. One thing that should be noted is that Hu's moments only cover 2D invariants. Impressions of 3D invariant images of an object viewed from different angles should be provided in the image database.

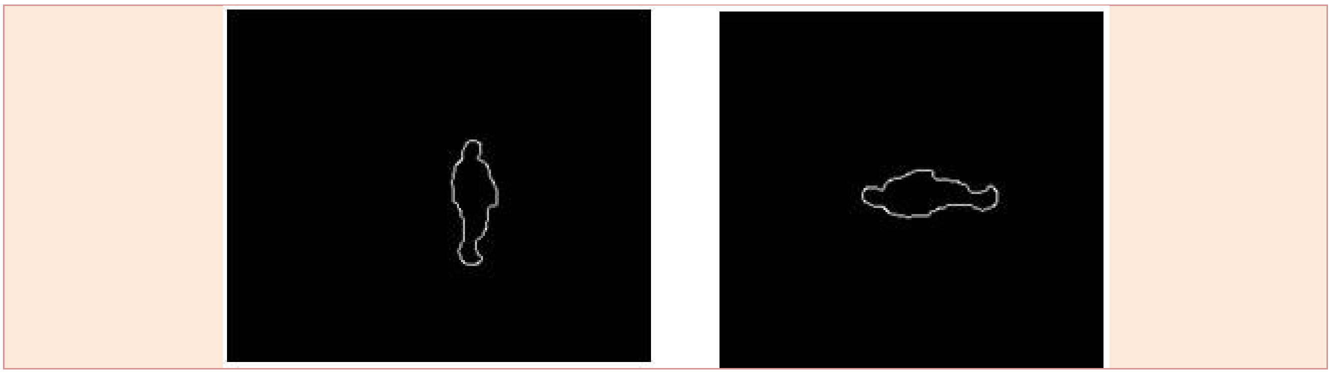


Figure 1. Original shape (left), rotated shape (right)

Since Hu moments are the earliest type of moments used in the object recognition field, they already have many applications in the real world. Mercimek et al. [3] in their experiment, tried to recognize three 3D objects. An object was rotated along the y-axis and photographed with every 5° rotation from 0° to 360°. They used some modification of Hu's moment invariant formula by introducing the distance between the object and the camera and the moment function oscillation radius. Using multi-layer perception with three output nodes, they classified the training data with 100% accuracy.

Table 1. Hu result example

	M1	M2	M3	M4	M5	M6	M7
Original	3.7841405	8.7173354	0.6875092	0.0230276	- 0.0028530	- 0.0543932	- 0.0063362
Rotated	3.7841405	8.7173354	0.6875092	0.0230276	- 0.0028530	- 0.0543932	0.0012996

Mao and Huang [4] used the moment eigenvector of a head-shoulder contour as the back-propagation neural network input for human identification by building a 2D model of the human head-and-shoulders. By adopting the partial contour human shape rather than whole features, they had better classification when solving the issue of the loss of property arising from human occluded easily in practical applications.

Zernike Moments

The two-dimensional Zernike moments, A_{nm} of order n with repetition m, of an image $f(\rho,\theta)$ are defined as:

$$A_{pq} = \frac{n+1}{\pi} \sum_{x=0}^{x=M-1} \sum_{y=0}^{y=N-1} f(\rho,\theta) V'_{pq}(\rho,\theta), \rho \leq 1 \tag{5}$$

where:

- (ρ, θ) is a polar coordinate,
- V'_{pq} is a complex conjugate,
- $\rho = \sqrt{x^2 + y^2}$ and $\theta = \arctan(y/x)$,
- V'_{pq} is a complex polynomial defined inside a unit circle with the formula:

$$V'_{pq}(\rho,\theta) = R_{pq}(\rho) \exp(jm\theta) \tag{6}$$

where:

- $\rho \leq 1$ and $j = \sqrt{-1}$ (imaginary unit).

$R_{pq}(\rho)$ is a radial polynomial, which can be generated using:

$$R_{pq}(\rho) = \sum_{s=0}^{n-|m|/2} (-1)^s \cdot \frac{(n-s)!}{s!\left(\frac{n+|m|}{2} - s\right)!\left(\frac{n-|m|}{2} - s\right)!} \cdot \rho^n - 2s \tag{7}$$

where:

- n is a positive integer, m can be a positive or negative integer,
- n - $|m|$ is even, $|m|$ <= n.

In order to reconstruct the original image $f(x, y)$ from the calculated Zernike moment features, this function is employed:

$$f'(x,y) = \sum_{p=0}^{N_{max}} \sum_{q=0} A_{pq} V_{pq}(\rho,\theta) \tag{8}$$

The magnitude of Zernike moments of a rotated image is similar to those before rotation [5]. Thus, $|A_{pq}|$ can be used as rotation invariant features of an image. However, Zernike moments are designed for rotation invariants only; in order to make translation and scaling invariants, image normalization needs to be performed.

To achieve translation normalization, the regular geometric moment of each image is used (m_{pq}). Translation invariance is achieved by transforming the image into a new one whose first order moments, m_{01} and m_{10}, are both equal to zero. This is done by transforming $f(x, y)$ into $f(x+x, y+y)$, where x and y are the image centroid point.

Scaling invariance is achieved by transforming the original image $f(x, y)$ into a new image $f(\alpha x, \alpha y)$, where $\alpha = \sqrt{\beta / m_{00}}$, β is a predetermined value, and m_{00} is the zero-th order moment of the original image, which is the object's area in a binary image.

An example of Zernike moment application is seen in Tripathy [6]. He developed an optical character recognition (OCR) system by first focusing only on reconstruction of the Indian Oriya alphabet using Zernike moments. It was shown that the reconstructions are quite similar to the original images using Zernike moments of order 30.

Image Database

The image database consists of two categories: testing and training. The training image consists of N class object types, each represented by one image. The training images are not varied for each object type because here we wanted to test the real capability of Hu moments and Zernike moments. Adding variations to the training images will help the algorithms and thus compensate for their inaccuracies.

The test images consist of $N \times V$ images, where V is the number of variations of each image class. These $N \times V$ images will undergo translation, scaling, and/or rotation transformation.

This project uses a grayscale Amsterdam Library of Object Images image database retrieved from <http://staff.science.uva.nl/~aloi/>. It is a collection of 1000 small objects recorded for scientific purposes.

From this database, two sample image databases are taken, each consisting of 10 and 50 images representing different classes of objects. Each image was resized to 144 x 144, preprocessed, and then each was subjected to affine transformations (translation, scaling, and rotation) to create nine new test images for each training image. Therefore, we obtained 90 images (from the sample of 10) and 450 images (from the sample of 50). The transformations are as follow:

- Rotated 30°, 140°, 250°, 325°.
- Translated 20 pixels on both axes.
- Scaled 1.2 times.
- Rotated 45° and translated -15 pixels on both axes.
- Rotated 45° and scaled 1.5 times.
- Rotated 50°, translated 10 pixels on both axes and scaled 1.3 times.

Classification

According to Friedman [7] there are six classifiers: decision functions, minimum-distance classifier, statistical approach, fuzzy, syntactic approach, and neural nets. The K-nearest-neighbor algorithm is in the minimum-distance classifier type.

Consider m classes C_j where (j = 1,2, ..., m) and a set of N sample patterns y_i where (i = 1,2, ..., N) with classification classes already known. Let x denote an input testing pattern. The nearest-neighbor classification approach classifies x in the class C_j if a y_i in $\min\|x - y_i\|$ for $1 <= i <= N$ belongs to C_j. The K-nearest-neighbor group K class with the closest distance from x and select the majority to be the class of x.

In our context, x is a 1-by-7 vector with the seven Hu moment invariant values as elements (in the case of Hu moments), or 1 by Feature_Max (in the case of Zernike moments). There will be N classification classes, and the value of K is determined by trial-and-error. Here, K is set to 1, because we only have 10 training images, each of which represents one class.

Experiment Analysis and Discussion

In this experiment, classification results from Hu moment invariants and Zernike moments are compared. For this comparison scheme, two results are seen from the two sample databases (10 classes and 50 classes), and the order of the Zernike moments will be varied. Also, a reconstructed image of each order will be given.

For the first sample database (10 classes), after the program was run, it was shown that the classification using Hu moments yielded 100% accuracy. Table 2 shows the classification results of using Zernike moments. For the second sample database (50 classes), Hu moments yielded 98% accuracy (9 errors). The results of the Zernike moments are shown in Table 3.

From these two observations, we can see that Zernike moments provide better accuracy than Hu moments as the order increases. These unfixed orders allow adaptability and flexibility of Zernike moments for a system (which is desirable), unlike the fixed order of Hu moments. Furthermore, other results that can be produced are reconstructions. Figure 2 shows the extracted Zernike moment features of two sample images.

After seeing the results of the classification and the reconstruction, we can see that lower-order Zernike moments contain the rough shape of the objects, and higher-order moments contain more detailed information.

Table 2. Classification results using Zernike moments (10 classes)

Order	Zernike Moment Errors	Accuracy (%)
5	12/90	86.67 %
6	4/90	95.56 %
7	1/90	98.89 %
8	0/90	100 %
9	0/90	100 %
10	0/90	100 %

Table 3. Classification results using Zernike moments (50 Classes)

Order	Zernike Moment Errors	Accuracy (%)
5	35	92.22 %
6	8	98.22 %

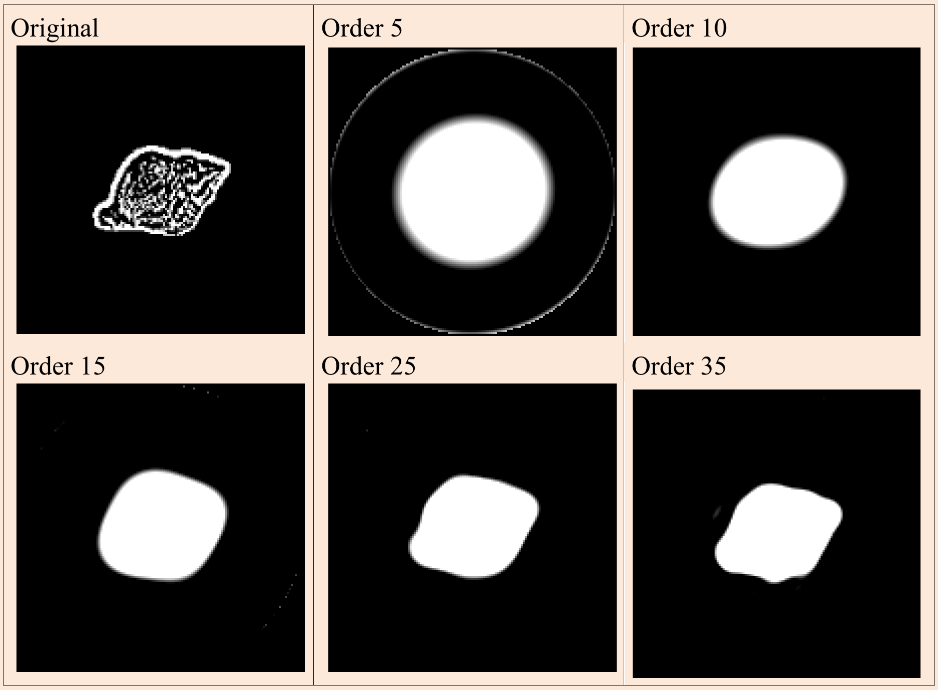


Figure 2. Extracted Zernike moment features

Conclusion

The objective of this project was achieved from seeing the experiment results, showing that Zernike moments are more accurate, flexible, and easier to reconstruct than Hu moments. The accuracy of Zernike moments is achieved by increasing the order of the moments. Flexibility means that we can choose the optimal order value for a system.

It is true that, in some cases, both Hu and Zernike moments achieved the same result. However, it has to be noted that the original image cannot be directly reconstructed from the Hu features. Even when we get the geometric moments, we have to transform it first into a Fourier domain.

References

[1] C. H. Teh, R. T. Chin, "Invariant image recognition by Zernike moments," *IEEE Transaction on Pattern Analaysis and Machine Intelligence.* vol. 10, no. 4, July 1988. Article (CrossRef Link)

[2] J. Flusser, T. Suk, B. Zitova, "Moments and moment invariants in pattern recognition," West Sussex: Wiley, 2009. Article (CrossRef Link)

[3] M. Mercimek, K. Gulez, T. V. Mumcu, "Real object recognition using moment invariants," *Sadhana*, 30, pp. 765-775, 2005. Article (CrossRef Link)

[4] Y. Mao, X. Huang, "Human recognition based on head-shoulder moment feature," in *Proc. of IEEE International Conference on Service Operations and Logistics, and Informatics*, pp.622-626, 2008. Article (CrossRef Link)

[5] A. Khotanzad, A. Y. Hong, "Invariant image recognition by Zernike moments," *IEEE Transaction on Pattern Analaysis and Machine Intelligence.* vol. 12, no. 5, May 1990. Article (CrossRef Link)

[6] J. Tripathy, "Reconstruction of Oriya alphabet using Zernike moments," *International Journal of Computer Applications* (0975 – 8887), vol. 8, no. 18, Oct. 2010. Article (CrossRef Link)

[7] M. Friedman, A. Kandel, "Introduction to pattern recognition: statistical, structural, neural and fuzzy logic approaches," London: Imperial College Press, 1999.

Appendix

(1)
$$M_{pq} = \iint_D p_{pq}(x,y) f(x,y)\,dxdy$$

(2)
$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\,dxdy$$

(3)
$$c_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x+iy)^p (x-iy)^q f(x,y)\,dxdy$$

(4)
$$\iint_\Omega p_{pq}(x,y)\cdot P_{mn}(x,y)\,dxdy = 0$$

(5)
$$\iint_\Omega w(x,y)\cdot p_{pq}(x,y)\cdot p_{mn}(x,y)\,dxdy = 0$$

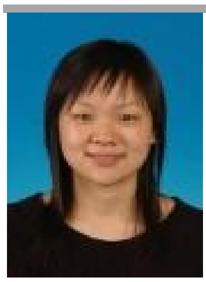
(6)
$$\begin{aligned}
M_1 &= (\eta_{20}+\eta_{02}),\\
M_2 &= (\eta_{20}-\eta_{02})^2 + 4\eta_{11}^2,\\
M_3 &= (\eta_{30}-3\eta_{12})^2 + (3\eta_{21}-\eta_{03})^2,\\
M_4 &= (\eta_{30}+\eta_{12})^2 + (\eta_{21}+\eta_{03})^2,\\
M_5 &= (\eta_{30}-3\eta_{12})^2(\eta_{30}+\eta_{12})\left[(\eta_{30}+\eta_{12})^2 - 3(\eta_{21}+\eta_{03})^2\right]\\
M_6 &= (\eta_{20}-\eta_{02})\left[(\eta_{30}+\eta_{12})^2 - (\eta_{21}+\eta_{03})^2\right] + 4\eta_{11}(\eta_{30}+\eta_{12})(\eta_{21}+\eta_{03}),
\end{aligned}$$

$$+(3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})\left[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2\right],$$

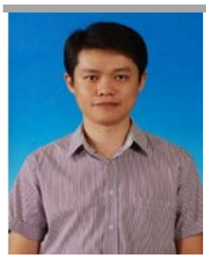
$$M_7=(3\eta_{21}-\eta_{03})(\eta_{30}+\eta_{12})\left[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2\right]$$

$$-(\eta_{30}+3\eta_{12})(\eta_{21}+\eta_{03})\left[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2\right],$$

Reza Kasyauqi Sabhara received his B. IT (Hons Artificial Intelligence) in 2011 from Multimedia University (MMU), Malaysia. He is currently working as a Software Engineer at Hitachi-Ebworx Sdn. Bhd. Malaysia. He is mainly interested in Artificial Intelligence.



Chin-Poo Lee received her Bachelor in Computer Science and MSc. in Information Technology, Multimedia University, Malaysia in 2004 and 2006, respectively. She is currently a PhD student of the Faculty of Information Science and Technology, Multimedia University, Malaysia. Her research interests include motion analysis, gait recognition and affective computing.



Kian-Ming Lim received his Bachelor of Information Technology and MEngSc. in 2004 and 2011, respectively. He is currently a PhD student of the Faculty of Information Science and Technology, Multimedia University, Malaysia. His research interests include machine learning, classification, and image processing.